

Programa del curso IC-1802

Introducción a la programación

Escuela de Computación

Carrera de Ingeniería de Computación, Plan 411.

I Parte. Aspectos relativos al plan de estudios

1 Datos generales

Nombre del curso: Introducción a la programación

Código: IC-1802

Tipo de curso: Teórico

Nº de créditos: 3

Nº horas de clase por semana: 3

Nº horas extraclase por semana: 6

Ubicación en el plan de estudios:

Requisitos: Ninguno

Correquisitos: IC-1803 Taller de programación

El curso es requisito de: IC-2001 Estructuras de Datos

Asistencia: Obligatoria

Suficiencia: No

Posibilidad de reconocimiento: Si

Vigencia del programa: I semestre 2022

2 Descripción General

Este curso comprende una introducción a los conceptos básicos para modelar y solucionar problemas de forma algorítmica así como las estrategias fundamentales para la construcción de programas de computadoras.

3 Objetivos

Objetivo General

Al finalizar el curso el estudiante debe ser capaz de desarrollar algoritmos básicos y realizarlos de manera sistemática en un modelo computacional específico o lenguaje de programación.

Objetivos Específicos

- Conocer algoritmos fundamentales de las ciencias de la computación.
- Especificar algoritmos por medio de abstracciones funcionales.
- Conocer y delimitar las etapas básicas asociadas en la construcción de algoritmos.
- Analizar la calidad, correctitud, eficiencia y mantenibilidad de una solución algorítmica.
- Conocer y utilizar los principales elementos semánticos y sintácticos de un modelo de programación concreto.

4 Contenidos

Resolución de problemas (4 semanas)

- Algoritmos
- Corrida Ejemplo
- Definición del problema
- Diseño del algoritmo
- Entradas, salidas, proceso y limitaciones
- Recursión
- Verificación y análisis del algoritmo
- Implementación del algoritmo
- Validación del programa

Algoritmos numéricos (4 semanas)

- Funciones matemáticas básicas
- Series
- Sucesiones
- Métodos numéricos
- Aritmética modular, números primos

Técnicas con listas (4 semanas)

- Iteración
- El concepto de lista
- Conjuntos implementados como listas
- Multiconjuntos
- Búsqueda
- Ordenamientos
- Relaciones y composiciones

Estructuras avanzadas (4 semanas)

- Árboles y su manipulación
- Árboles binarios ordenados
- Variables imperativas
- Funciones como datos
- Manejo imperativo de datos
- Manejo básico de objetos

II Parte. Aspectos operativos

5 Metodología de enseñanza y aprendizaje

Se emplearán técnicas de explicación por parte del profesor utilizando la herramienta zoom, en donde se desarrollarán los aspectos teóricos y prácticos más relevantes de los diferentes temas. Además se combinarán con una alta participación por parte de los estudiantes, utilizando la cámara y durante el transcurso de las lecciones, se realizarán llamadas orales, respuestas a casos y de trabajos en grupo.

Se presupone que el alumno profundiza los temas abordados en la clase en las lecturas recomendadas por el profesor y que el estudiante será responsable de desarrollar los proyectos programados que se asignen en el curso.

6 Evaluación

El curso será evaluado mediante dos rubros:

Exámenes parciales (45%): Se realizarán 3 (tres) exámenes parciales durante el semestre, cada uno de 15%, programados tentativamente para las semanas: 8, 12 y 16.

Tareas Cortas (40%): Durante el semestre se asignarán tareas cortas a medida que se presenten los contenidos en el curso. Las primeras dos tareas consisten en presentar el **perfilPersonal** de cada estudiante y asistir a la proyección de las películas **Stand and Deliver** y **Coach Carter** (independientemente si las han visto). Las otras ocho tareas consisten en la realización de los ejercicios del libro “Fundamentos de Programación” del profesor, estos ejercicios se realizarán en parejas. Estas tareas deben entregarse cada 2 semanas de clases, a menos que el profesor posponga explícitamente alguna entrega.

Detalle de las tareas por capítulo (a partir del capítulo 3 se debe utilizar python en todos los entregables):

Capítulo 1: Sección 1.8.1, Sección 1.8.3, Sección 1.8.4 (semana 3)

Capítulo 2: Sección 2.11.2 (semana 5)

Capítulo 3: Sección 3.7.2, Sección 3.7.3 (semana 7)

Capítulo 4: Sección 4.3.1, Sección 4.3.2, Sección 4.3.3, Sección 4.3.5 (semana 9)

Capítulo 5: Sección 5.5.1, Sección 5.5.3, Sección 5.5.4 (semana 11)

Capítulo 6: Sección 6.6.1, Sección 6.6.2, Sección 6.6.3 (semana 13)

Capítulo 7: Sección 7.6.2 (semana 15)

Capítulo 8: Escoger algún ejercicio no realizado en clase del capítulo y hacerlo (lunes semana 18)

Quices (15%): Durante el semestre se realizará un quiz (1 % c/uno) por semana a partir de la semana 2, los cuales servirán de diagnóstico para verificar la comprensión de los temas vistos. Es importante que los estudiantes corrijan cualquier error en un quiz porque la materia es acumulativa.

	%
Pruebas cortas y tareas	40
Exámenes	45
Quices	15
	100

El contenido académico de las actividades son acumulativos.

El porcentaje de las pruebas cortas y tareas será el promedio de las notas obtenidas en cada una de ellas.

En caso de que se detecte un plagio o intento de fraude en cualquier asignación por parte de un estudiante se procederá según la reglamentación del Instituto Tecnológico de Costa Rica.

Todas las tareas deben entregarse a los correos edramirez@itcr.ac.cr ó ed.rojas@itcr.ac.cr con el asunto **IC-1802 Intro - Ejercicios cap x**, donde x representa el número de capítulo que se está resolviendo. Si hubiera algún ejercicio mal resuelto o incompleto, el estudiante debe solicitar una cita de consulta para aclarar cuál fue el error, de modo que aprenda correctamente como resolverlo.

La forma de entrega de los ejercicios debe ser en LaTeX, pero puede realizarse el PDF añadiendo imágenes escaneadas si el ejercicio así lo requiere, salvo que sea código fuente. Debe indicarse en el documento el número de ejercicio que se está resolviendo.

III Parte. Aspectos Administrativos

7 Canales de Comunicación

1. Cualquier **correo** que sea enviado al asistente o al profesor **directamente** deberá tener el siguiente formato. **Subject:** IC-1802 Intro - [asunto], donde el asunto corresponde al tema de la consulta. No se incluyen los paréntesis cuadrados. **Cuerpo:** Se debe iniciar el cuerpo con el nombre completo y carnet de cada estudiante (en el caso que sea en grupos, se debe colocar el nombre de todos junto con su carnet)
2. Cuando se trabaja en grupos y se envía un correo, el mismo deberá tener **copia** hacia todos los integrantes del grupo.
3. En caso de que se agreguen archivos adjuntos al correo, se debe poner una **descripción** de cada archivo adjunto, junto con sus contenidos.
4. Toda **consulta** debe ser **colectiva** y para ello se cuenta con el grupo de Telegram.

8 Entrega de Asignaciones

1. **Todos** los trabajos **escritos** deben de presentarse **digitalmente** y si no tienen bibliografía, se otorgará una nota de cero.
2. En caso de entrega **tardía** de una asignación, se incluye una **multa de 1.5 puntos por hora**, luego de vencido el plazo previamente establecido.
3. El profesor se guarda el derecho de **revisar la ortografía**, redacción y coherencia con puntos negativos, en TODAS las asignaciones, llámense quices, exámenes o tareas. Al menos se otorgará un punto negativo por cada falta. Salvo que se confundan cualesquiera de estas palabras, eso implicaría 5 puntos: hay, ahí, ¡Ay! o se escriban mal.
4. En las **revisiones presenciales virtuales** deben estar presentes todos los miembros del equipo.
5. Los resúmenes o ensayos deberán de presentarse al **séptimo día** antes de la clase posterior a su asignación.
6. Los proyectos o tareas cortas que contengan **más de 1 archivo** deberán de estar comprimidos utilizando el formato: **zip** salvo que se acuerde lo contrario en el grupo de Telegram o en clase.
7. Todas las asignaciones escritas, por ejemplo, tareas ensayos, documentaciones y resúmenes, deben de presentarse en formato **pdf**. Y se deben presentar **SIN comprimir** y haberse realizado en LaTeX.
8. Se debe de realizar la entrega de TODOS los proyectos y exámenes asignados que posean un valor mayor o igual a un 10%, de lo contrario se considerará un abandono de curso (según reglamento del TEC).
9. Las pruebas cortas pueden consistir en una asignación para trabajo fuera de clase.

9 Aspectos Adicionales

1. Cualquier hora dada en el curso es en **formato 24h**.
2. En cualquier fecha que el profesor asigne se puede asumir que es para el **presente año**.
3. El huso horario del curso oficial es **GMT-6**.
4. Queda agendado un **quiz** en cada una de las lecciones del curso, el profesor tomará la decisión de aplicar o no el quiz.
5. El profesor puede evaluar temas generales o específicos que pertenezcan a otros cursos y que se consideren relevantes para el presente curso (principalmente requisitos o matemática discreta).
6. El curso se aprueba con nota de **70**.
7. No hay examen de reposición.
8. **No es posible tomar fotografías**, videos o sonidos dentro de la clase (para eso lo hago yo en pandemia y lo subo al canal de Telegram).
9. Los exámenes **no se reponen** y se deben entregar a tiempo.
10. El contenido académico de las actividades, llámense pruebas cortas, tareas o proyectos, es **acumulativo**.
11. Los **fraudes** en cualquier actividad llevada a cabo durante el semestre implicará que se perderá el curso y se reportará la nota mínima. Además se enviará una carta al expediente del estudiante. No se tolerará ningún fraude de ninguna forma en este curso.
12. La asistencia a clases es obligatoria. Si un estudiante falta al 15% o más del total de horas semestrales pierde automáticamente el curso. Se pasará lista a las 7:55 am, si no se está presente, tiene 3 días hábiles para justificar su ausencia vía correo electrónico, el profesor determinará si es una excusa válida.
13. Lo cubierto en tareas cortas, lecturas, proyectos es **materia de examen**.
14. El medio oficial de comunicación es el grupo de Telegram del grupo, cada estudiante debe de revisarlo al menos una vez al día.

10 Bibliografía

Se utilizará como libro de texto principal, “Fundamentos de Programación” del profesor Eddy Ramírez:

- Helo, J. “Introducción a la Programación con Scheme”, Editorial Tecnológica de Costa Rica, Cartago, 2000.
- Friedaman, D; Felleisen, M. “The Little Schemer”, IV Ed, Science Research Associates Inc, SRA Inc, The MIT Press, 1999.
- Abelson, H; Sussman, G. “Structure and Interpretation of Computer Programs”, The MIT Press, McGraw Hill Company, Massachusetts, 1985.
- Downey Allen, Elkner Jeffrey, Meyers Chris “How to think like a computer scientist”. Grean Tea Press. 2002

- Solano Jaime “Introducción a la programación en Python”. Editorial tecnológica.

11 Profesor

Eddy Ramírez Jiménez, MSc. - Edgar Rojas Muñoz, PhD

Email: edramirez@itcr.ac.cr - ed.rojas@itcr.ac.cr

Página: <http://www.ic-itcr.ac.cr/~edramirez>

Twitter: @eddituss - @EdkazarJarom

Horario y lugar de consulta: Lunes 9:00 am - 12:00 md. Telegram en @eddituss y @Edkazar

PerfilEddy.pdf - PerfilEdgar.pdf